

Claims:

What is claimed is:

1. A method for efficiently dispatching threads awaiting messages in a multi-threaded communication library comprising:
 - preassigning threads to messages to be received;
 - putting to sleep, those threads whose assigned messages have not been received;
 - upon receipt of a message, awakening its preassigned thread; and
 - executing said awakened thread, thereby processing the received message.
2. The method of claim 1 wherein the selection of the thread to be dispatched is based on its priority as set when the thread is put to sleep.
3. The method of claim 1 wherein said preassigning threads step comprises:
 - creating a thread-specific structure for each thread, each thread-specific structure having a ready flag and a condition variable unique to its preassigned thread;
 - creating a handle for each message to be received; and
 - having a thread invoke message passing logic for a particular handle, thereby associating the thread and the message.
4. The method of claim 3 wherein said putting to sleep step comprises:
 - enqueueing for a received message, a preassigned thread-specific structure into a first queue;
 - writing into said handle associated with the message received, an identification of said thread-specific structure enqueued for the received message, and
 - placing said thread-specific structure for the received message in the WAIT condition.

1 5. The method of claim 4 wherein said awakening step comprises;
2 completing said received message;
3 changing the condition of the thread-specific structure for the completed received
4 structure to the READY condition; and
5 dequeuing with a queue manager, the next thread-specific structure in said first
6 queue in the READY condition and sending its thread a thread awakening condition
signal.

1 6. The method of claim 5 further comprising;
2 allocating in said preassigning step, buffer space for storing messages to be
3 received; and
4 in said putting to sleep step, identifying in said handle the buffer in which the
5 message associated with the handle is to be stored when it is received.

1 7. The method of claim 6 wherein said completing said received message comprises
2 storing said received message in the buffer identified in the associated handle for the
3 received message.

1 8. The method of claim 5 wherein said queue manager dequeues the next
2 thread-specific structure using a First-In-First-Out policy.

1 9. The method of claim 5 wherein said queue manager dequeues the next
2 thread-specific structure using a Last-In-First-Out policy.

1 10. The method of claim 5 wherein said queue manager dequeues the next
2 thread-specific structure based on a priority value contained in said structure.

1 11. The method of claim 5 further comprising obtaining a lock for the handle
2 associated with said received message such that the awakened thread may process only the
3 received message.

1 12. The method of claim 11 further comprising releasing said lock after said awakened
2 thread has processed said received message such that said awakened thread may continue
3 with other work.

1 13. A computer program product comprising a computer useable medium having
2 computer readable program code means therein for efficiently dispatching threads
3 awaiting messages in a multi-threaded communication library, said computer readable
4 program code means in said computer program product comprising:
5 computer readable program code means for preassigning threads to messages to be
6 received;
7 computer readable program code means for putting to sleep, those threads whose
8 assigned messages have not been received;
9 computer readable program code means for, upon receipt of a message, awakening
10 its preassigned thread; and
11 computer readable program code means for executing said awakened thread,
12 thereby processing the received message.

1 14. The computer program product of claim 13 wherein the selection of the thread to
2 be dispatched is based on its priority as set when the thread is put to sleep.

1 15. The computer program product of claim 13 wherein said computer readable
2 program code means for preassigning threads comprises:
3 computer readable program code means for creating a thread-specific structure for
4 each thread, each thread-specific structure having a ready flag and a condition variable
5 unique to its preassigned thread;
6 computer readable program code means for creating a handle for each message to
7 be received; and
8 computer readable program code means for having a thread invoke message
9 passing logic for a particular handle, thereby associating the thread and the message.

1 16. The computer program product of claim 15 wherein said computer readable
2 program code means for putting to sleep comprises:

3 computer readable program code means for enqueueing for a received message, a
4 preassigned thread-specific structure into a first queue;

5 computer readable program code means for writing into said handle associated
6 with the message received, an identification of said thread-specific structure enqueued for
7 the received message, and

8 computer readable program code means for placing said thread-specific structure
9 for the received message in the WAIT condition.

1 17. The computer program product of claim 16 wherein said computer readable
2 program code means for awakening comprises;

3 computer readable program code means for completing said received message;

4 computer readable program code means for changing the condition of the
5 thread-specific structure for the completed received structure to the READY condition;
6 and

7 computer readable program code means for dequeuing with a queue manager, the
8 next thread-specific structure in said first queue in the READY condition and sending its
9 thread a thread awakening condition signal.

1 18. The computer program product of claim 17 further comprising;

2 computer readable program code means for allocating in said preassigning step,
3 buffer space for storing messages to be received; and

4 said computer readable program code means for putting to sleep includes,
5 computer readable program code means for identifying in said handle the buffer in which
6 the message associated with the handle is to be stored when it is received.

1 19. The computer program product of claim 18 wherein said computer readable

2 program code means for completing said received message comprises computer readable

3 program code means for storing said received message in the buffer identified in the
4 associated handle for the received message.

1 20. The computer program product of claim 17 wherein said queue manager includes
2 computer readable program code means for dequeueing the next thread-specific structure
3 using a First-In-First-Out policy.

1 21. The computer program product of claim 17 wherein said queue manager includes
2 computer readable program code means for dequeueing the next thread-specific structure
3 using a Last-In-First-Out policy.

1 22. The computer program product of claim 17 wherein said queue manager includes
2 computer readable program code means for dequeueing the next thread-specific structure
3 based on a priority value contained in said structure.

1 23. The computer program product of claim 17 further comprising computer readable
2 program code means for obtaining a lock for the handle associated with said received
3 message such that the awakened thread may process only the received message.

1 24. The computer program product of claim 23 further comprising computer readable
2 program code means for releasing said lock after said awakened thread has processed said
3 received message such that said awakened thread may continue with other work.

1 25. An apparatus for efficiently dispatching threads awaiting messages in a
2 multi-threaded communication library comprising:
3 means for preassigning threads to messages to be received;
4 means for putting to sleep, those threads whose assigned messages have not been
5 received;
6 means for, upon receipt of a message, awakening its preassigned thread; and
7 executing said awakened thread, thereby processing the received message.

1 26. The apparatus of claim 25 wherein the selection of the thread to be dispatched is
2 based on its priority as set when the thread is put to sleep.

1 27. The apparatus of claim 25 wherein said means for preassigning threads comprises:
2 means for creating a thread-specific structure for each thread, each thread-specific
3 structure having a ready flag and a condition variable unique to its preassigned thread;
4 means for creating a handle for each message to be received; and
5 means for having a thread invoke message passing logic for a particular handle,
6 thereby associating the thread and the message.

1 28. The apparatus of claim 27 wherein said means for putting to sleep comprises:
2 means for enqueueing for a received message, a preassigned thread-specific structure
3 into a first queue;
4 means for writing into said handle associated with the message received, an
5 identification of said thread-specific structure enqueued for the received message, and
6 means for placing said thread-specific structure for the received message in the
7 WAIT condition.

1 29. The apparatus of claim 28 wherein said means for awakening comprises;
2 means for completing said received message;
3 means for changing the condition of the thread-specific structure for the completed
4 received structure to the READY condition; and
5 means for dequeuing with a queue manager, the next thread-specific structure in
6 said first queue in the READY condition and sending its thread a thread awakening
7 condition signal.

1 30. The apparatus of claim 29 further comprising;
2 means for allocating in said preassigning step, buffer space for storing messages to
3 be received; and

4 in said means for putting to sleep, means for identifying in said handle the buffer
5 in which the message associated with the handle is to be stored when it is received.

1 31. The apparatus of claim 30 wherein said means for completing said received
2 message comprises means for storing said received message in the buffer identified in the
3 associated handle for the received message.

1 32. The apparatus of claim 29 wherein said queue manager includes means for
2 dequeuing the next thread-specific structure using a First-In-First-Out policy.

1 33. The apparatus of claim 29 wherein said queue manager includes means for
2 dequeuing the next thread-specific structure using a Last-In-First-Out policy.

1 34. The apparatus of claim 29, wherein said queue manager includes means for
2 dequeuing the next thread-specific structure based on a priority value contained in said
3 structure.

1 35. The apparatus of claim 29 further comprising means for obtaining a lock for the
2 handle associated with said received message such that the awakened thread may process
3 only the received message.

1 36. The apparatus of claim 35 further comprising means for releasing said lock after
2 said awakened thread has processed said received message such that said awakened thread
3 may continue with other work.

1 37. An apparatus comprising:
2 a data processing system;
3 a multi-threaded communication library in said data processing system;
4 a thread dispatcher in said data processing system for efficiently dispatching
5 threads awaiting messages in said multi-threaded communication library;

6 computer code which preassigns threads to messages to be received;
 7 computer code which puts to sleep those threads whose assigned messages have
 8 not been received;
 9 computer code which, upon receipt of a message, awakens its preassigned thread;
 10 and
 11 computer code which executes said awakened thread, thereby processing the
 12 received message.

1 38. The apparatus of claim 37 wherein the selection of the thread to be dispatched is
 2 based on its priority as set when the thread is put to sleep.

1 39. The apparatus of claim 37 wherein said computer code which preassigns threads
 2 comprises:

3 computer code which creates a thread-specific structure for each thread, each
 4 thread-specific structure having a ready flag and a condition variable unique to its
 5 preassigned thread;

6 computer code which creates a handle for each message to be received; and

7 computer code which causes a thread invoke message passing logic for a particular
 8 handle, thereby associating the thread and the message.

1 40. The apparatus of claim 39 wherein said computer code which puts to sleep
 2 comprises:

3 computer code which enqueues for a received message, a preassigned
 4 thread-specific structure into a first queue;

5 computer code which writes into said handle associated with the message received,
 6 an identification of said thread-specific structure enqueued for the received message, and

7 computer code which places said thread-specific structure for the received message
 8 in the WAIT condition.

1 41. The apparatus of claim 40 wherein said computer code which awakens comprises;
2 computer code which completes said received message;
3 computer code which changes the condition of the thread-specific structure for the
4 completed received structure to the READY condition; and
5 computer code which dequeues with a queue manager, the next thread-specific
6 structure in said first queue in the READY condition and sending its thread a thread
7 awakening condition signal.

1 42. The apparatus of claim 41 further comprising;
2 in said computer code which preassigns, computer code which allocates buffer
3 space for storing messages to be received; and
4 in said computer code which puts to sleep, computer code which identifies in said handle
5 the buffer in which the message associated with the handle is to be stored when it is
6 received.

1 43. The apparatus of claim 42 wherein said computer code which completes said
2 received message comprises computer code which stores said received message in the
3 buffer identified in the associated handle for the received message.

1 44. The apparatus of claim 41 wherein said queue manager includes computer code
2 which dequeues the next thread-specific structure using a First-In-First-Out policy.

1 45. The apparatus of claim 41 wherein said queue manager includes computer code
2 which dequeues the next thread-specific structure using a Last-In-First-Out policy.

1 46. The apparatus of claim 41 wherein said queue manager includes computer code
2 which dequeues the next thread-specific structure based on a priority value contained in
3 said structure.

1 48. The apparatus of claim 47 further comprising computer code which releases said
2 lock after said awakened thread has processed said received message such that said
3 awakened thread may continue with other work.

04894E 000000